

Datenbanken und Datensicherheit

Übungsaufgaben 5 – 13. November 2024

Hinweis: Es ist vorgesehen, daß Sie zur Lösung dieser Aufgaben Web-Recherchen betreiben. Dies wird während der Klausur nicht möglich sein. In diesem Sinne sind diese Aufgaben *keine* typischen Klausuraufgaben.

Aufgabe 1: Text in Bild

Die folgenden Unix-Kommandos verstecken den Text `aufgabe-1-1.txt` in der Bilddatei `aufgabe-1-1.jpg`:

```
$ ls -l 320px-1-month-old_kittens_32.jpg
-rw-r--r-- 1 peter peter 15042 16. Jun 2015 320px-1-month-old_kittens_32.jpg
$ display 320px-1-month-old_kittens_32.jpg
```



```
$ jpegtran 320px-1-month-old_kittens_32.jpg > aufgabe-1-1.jpg
$ ls -l aufgabe-1-1.jpg
-rw-r--r-- 1 peter peter 12457 12. Nov 22:32 aufgabe-1-1.jpg
$ display aufgabe-1-1.jpg
```



```
$ cat aufgabe-1-1.txt
Süüüß! :-)
$ cat aufgabe-1-1.jpg aufgabe-1-1.txt > aufgabe-1-2.jpg
$ ls -l aufgabe-1-2.jpg
-rw-r--r-- 1 peter peter 12472 12. Nov 22:33 aufgabe-1-2.jpg
$ display aufgabe-1-2.jpg
```



```
$ dd status=none if=aufgabe-1-2.jpg bs=1 \
skip=$(jpegtran aufgabe-1-2.jpg | wc -c) > aufgabe-1-2.txt
$ cat aufgabe-1-2.txt
Süüüß! :-)
```

- (a) Wie funktioniert das Verstecken, und wieso läßt sich das Bild weiterhin anzeigen?
- (b) Wie kann ein Angreifer, der das Originalbild nicht kennt, trotzdem erkennen, daß in dem Bild zusätzliche Informationen versteckt wurden, und diese extrahieren?
- (c) Welchen Zweck erfüllt der erste Aufruf von `jpegtran`?
Was ändert sich, wenn man diesen Aufruf wegläßt?

Bildquelle: https://commons.wikimedia.org/wiki/File:1-month-old_kittens_32.jpg
Verwendung gemäß Lizenz: CC BY-SA 2.0 Generic

Aufgabe 2: Text in ausführbarer Datei

Da in Shell-Skripten (oder anderen Skriptsprachen) gespeicherte Passwörter direkt sichtbar sind, erfolgt die Passwort-Abfrage in diesem Beispiel durch ein kompiliertes Programm. Der C-Quelltext ([aufgabe-2.c](#)) lautet:

```
#include <stdio.h>
#include <unistd.h>
#include <string.h>

int main (void)
{
    char *password = getpass ("Password:_");
    if (strcmp (password, "gehe1m!!1") == 0)
        printf ("You_have_access.\n");
    else
        printf ("Wrong_password,_sorry.\n");
    return 0;
}
```

Finden Sie einen Weg, das Passwort aus der ausführbaren Datei ([aufgabe-2](#)) zu extrahieren.

Aufgabe 3: Programme installieren

Zur Installation der Programme (a) bis (d) finden Sie jeweils die folgende Dokumentation:

- (a) „Hängen Sie die Zeile `deb https://example.com/repository example main` an die Datei `/etc/apt/sources.list` an, und rufen Sie anschließend `apt update` und danach `apt install example` auf.“
- (b) „Geben Sie den Befehl `curl https://example.com/install.sh | sudo sh` ein.“
- (c) „Laden Sie die Datei `https://example.com/example.deb` herunter, und geben Sie danach den Befehl `dpkg -i example.deb` ein.“
- (d) „Installieren Sie das Programm über den Paket-Manager Ihrer Distribution, indem Sie den Befehl `apt install example` eingeben.“

Wie kann ein Angreifer, der die Webseiten unter [example.com](#) manipulieren kann, Befehle auf dem Rechner derjenigen Person ausführen, die die Software installiert und anschließend benutzt? Wie groß ist jeweils der Aufwand für den Angreifer?

Viel Erfolg!